valentin
ETIKETTENDRUCKSYSTEME

# User guide

Printer driver
Version 2.1.x
Microsoft® Windows

September 2010

Copyright by Carl Valentin GmbH / 79820550710

Information on the scope of delivery, appearance, performance, dimensions and weight reflect our knowledge at the time of printing.

We reserve the rights to make modifications.

All rights, including those regarding the translation, are reserved.

No part of this document may be reproduced in any form (print, photocopy or any other method) or edited, copied or distributed electronically without written permission from Carl Valentin GmbH.

**Certified driver**

Version 2.1 Build 7 of printer driver is certified for Windows XP®, Vista® und 7® (x386 and x64) and fulfils the high quality standards of Microsoft.

Windows Vista and the Windows logo are trademarks of the Microsoft group of companies.

Microsoft, Windows, and the Windows logo are trademarks of the Microsoft group of companies.

valentin
ETIKETTENDRUCKSYSTEME

**Carl Valentin GmbH**

Postfach 3744 D-78026 Villingen-Schwenningen
Neckarstraße 78-82+94 D-78056 Villingen-Schwenningen

Phone: +49 7720 9712-0
Fax: +49 7720 9712-9901
E-mail: info@valentin-carl.de
**www.valentin-carl.de**

# Table of contents

**Table of contents ... 3**

**1 Installation of printer driver ... 5**
1.1 Preparations ... 5
1.2 Installation ... 7
1.3 Import - Export ... 12

**2 Application of printer commands ... 13**
2.1 Explanation ... 13
2.2 Formatting of text ... 13
2.3 Printer commands ... 18

**3 Application of 2D bar codes ... 21**
3.1 Explanation ... 21

**4 Application of variables ... 21**
4.1 Definitions ... 21
4.2 Counter ... 22
4.3 Date/time ... 22
4.4 Using in document ... 23
4.5 Using wildcards ... 24
4.6 Defining wildcards ... 25
4.7 Label example ... 28

**5 Error handling ... 29**
5.1 Recognising problems ... 29
5.2 Protocol files ... 30
5.3 Possible errors ... 32

**6 Index ... 35**

# 1 Installation of printer driver

## 1.1 Preparations

**Deleting the old driver**

Before installing a new printer driver, the before used driver must be deleted and removed from server. The documentation shows the manual steps which are to be executed. The installation program offers a function with which deleting of driver is effected automatically from the server properties (see page 7).

**Deleting using Windows XP®**

Click **Start**, **Control Panel** and **Printers and Faxes**.

Click the right mouse button on the driver you want to delete and then click **Delete**.

From the **File** menu select **Server Properties**.

Select the **Drivers** tab, mark the driver you want to delete and click **Remove**.

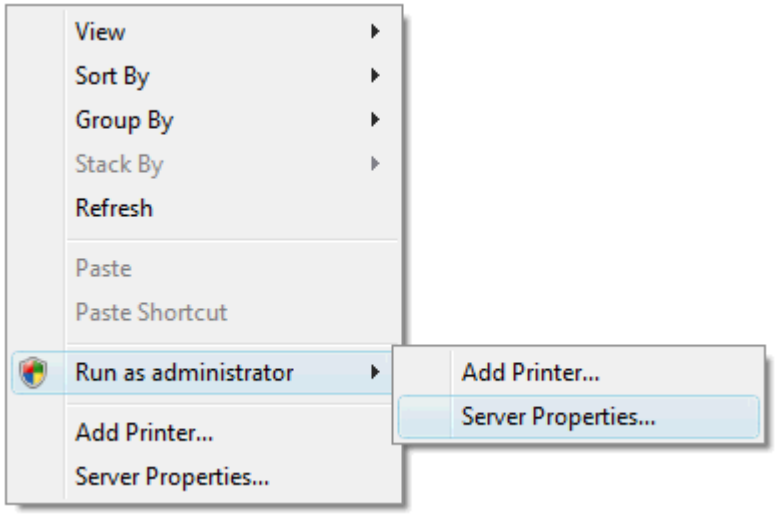
**Deleting using Windows Vista®**

Click **Start**, **Control Panel** and **Printers**.

Click the right mouse button on the driver you want to delete and then click **Delete**.

Before the driver can be removed in the server properties, the function **Run as administrator** must be selected.

Click the right mouse button inside the list of available printers.




Select **Run as administrator** and then **Server Properties**.

Click **Continue** to start the procedure.

Select the **Drivers** tab, mark the driver you want to delete and click **Remove**.

Select the function **Remove driver and driver package** and then follow the instructions on the screen.

**Deleting using Windows 7®**

Click **Start**, **Devices and Printers**.

Click the right mouse button on the driver you want to delete.

From the menu bar click **Remove device**.

Click a printer you want and afterwards the menu item **Print server properties** is shown in the menu bar.

Click the **Printer server properties** item.

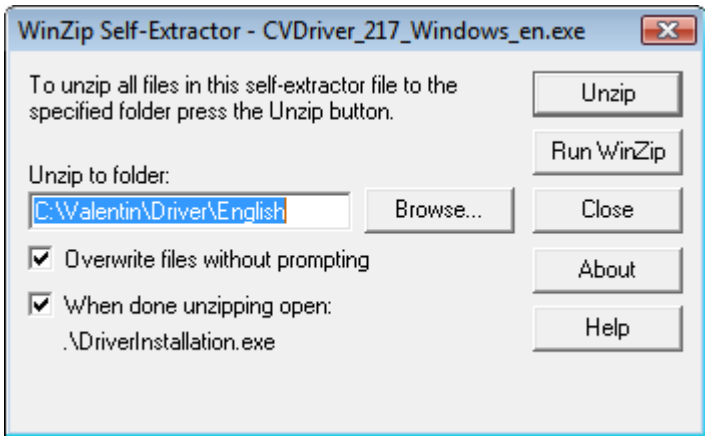
From the **Driver** tab select the driver you want to delete and then click **Remove**.

**Extracting printer files**

Current driver versions can be downloaded free of charge from the appropriate area on our web site.

Save the selected exe file on your computer and start the file.

Unless otherwise indicated, the files are saved in the directory **C:\Valentin\Driver/English**.




If the installation is to be effected under Windows XP® or Vista® per Plug & Play then deselect the **When done unzipping open**. Afterwards click **Unzip**.

## 1.2 Installation

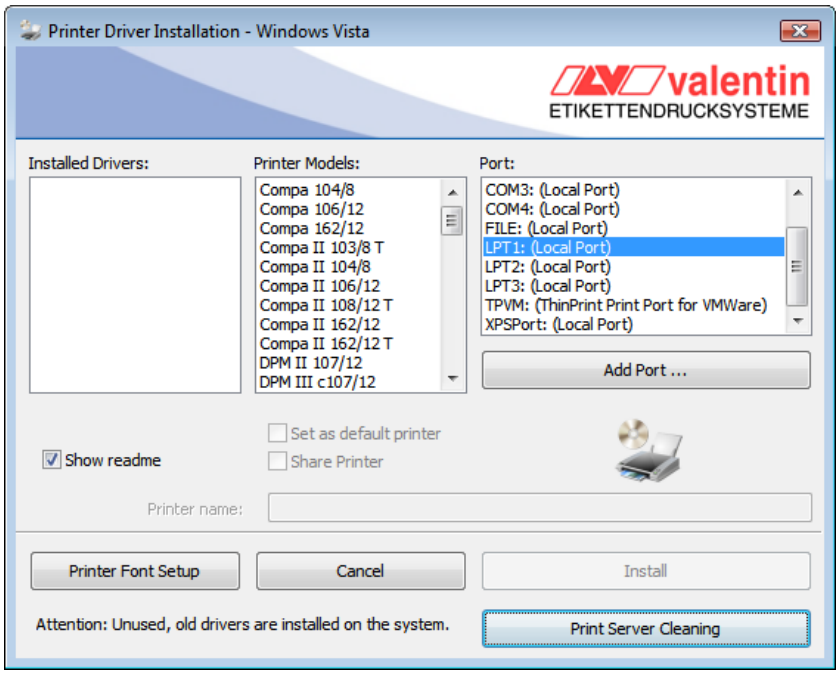
There are different possibilities for installing the printer available. The simplest version is the use of the included installation program.

**Automatic cleaning of print server**

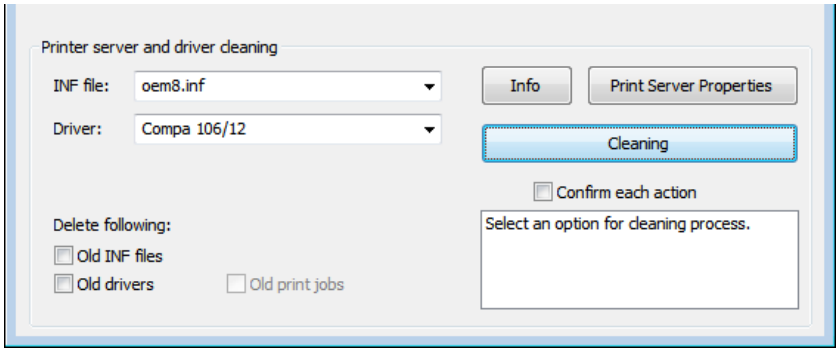
The Printer driver installation dialog offers a function for an automatic cleaning of print server. No longer used driver and installation files can be removed from program.

Start the installation program.

If older installations are existing, click **Clean print server**.




The dialog appears its extension.




Select all check boxes under **Delete following items** and then click the **Cleaning** button. The selection of **Confirm each action** can be deleted if desired.

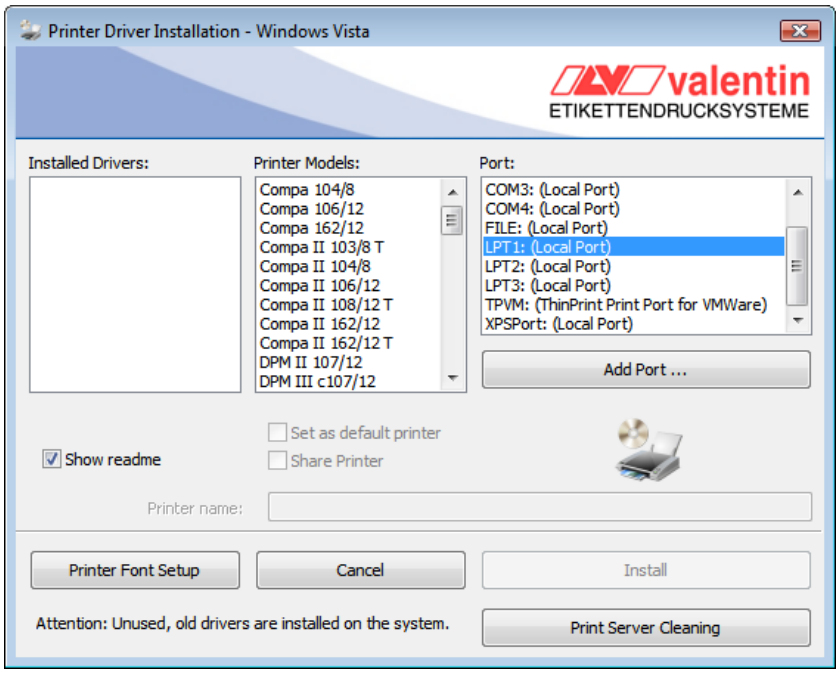
The installation can be continued.

**Installation with Setup**

Extract the driver files (description see page 6) and select in the corresponding box the check box **When done unzipping open**.

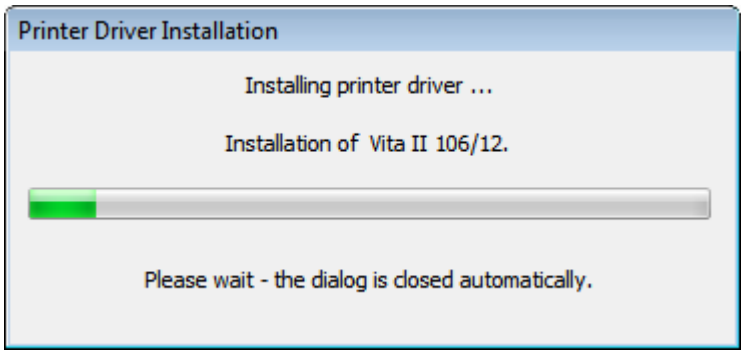
If you have a printer driver installation CD, insert the CD and run the file **Start.exe**.
Click **Printer driver** and select your operating system.

The **Printer Driver Installation** dialog box appears.

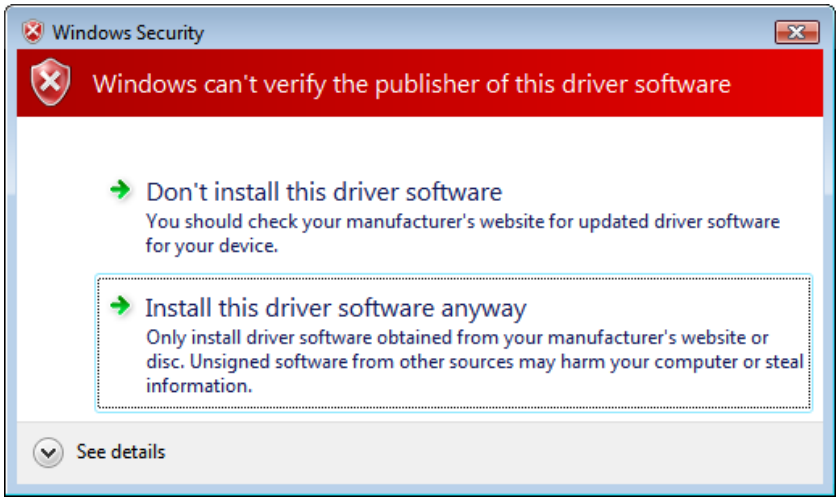



Select the desired printer and the appropriate port.

Click the **Install** button to start the procedure.


Printer Driver Installation
Installing printer driver ...
Installation of Vita II 106/12.
Please wait - the dialog is closed automatically.

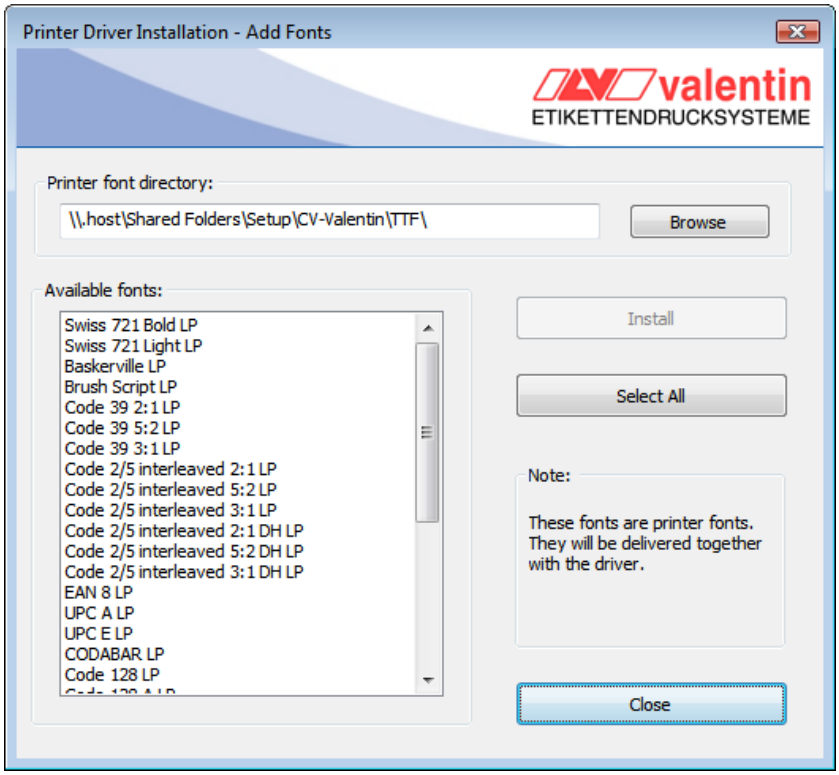
If a uncertified version is installed the following warning message appears after a few seconds:


Windows Security
Windows can't verify the publisher of this driver software
Don't install this driver software
You should check your manufacturer's website for updated driver software for your device.
Install this driver software anyway
Only install driver software obtained from your manufacturer's website or disc. Unsigned software from other sources may harm your computer or steal information.
See details

Confirm the note with **Install this driver software anyway**.

This has no reaction to the functionality of driver. However, this driver fulfills the quality conditions of Microsoft.

If not all True Type fonts available in the printer were installed yet the following dialog appears at the end of installation:




Select the desired font and click the **Install** button.

Afterwards click **Close**.

Now the driver can be set up by the **Properties** dialog box.

The logo test is inapplicable for the driver versions 2.1.4. and 2.1.7 because this versions are certified.

**Installation with Plug & Play**

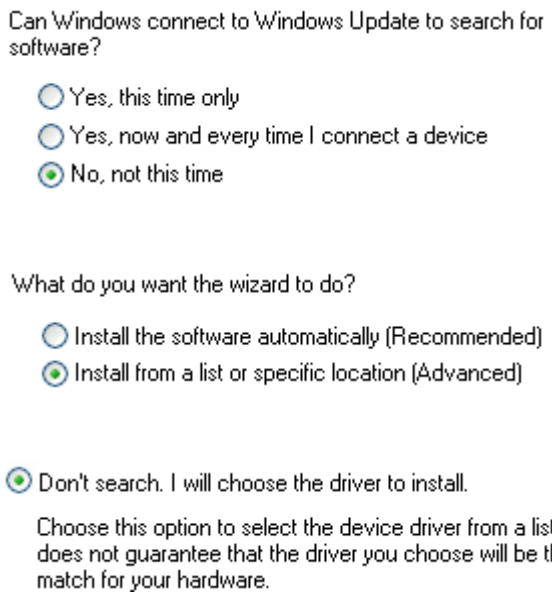
Plug & Play enables the computer to recognise a device automatically, to configure it and to install the appropriate printer driver.

For initial operation of printer using the parallel or USB interface, we recommend the installation of printer driver by means of the Plug & Play function. Otherwise the **Found New Hardware Wizard** appears at each re-start of Windows®.

After connecting and switching on the printer, the message **New hardware found** appears down right on the screen.

Click the mouse pointer to any position inside the field and the **Found New Hardware Wizard** appears.

Select the check boxes as displayed below:


Can Windows connect to Windows Update to search for software?
Yes, this time only
Yes, now and every time I connect a device
No, not this time
What do you want the wizard to do?
Install the software automatically (Recommended)
Install from a list or specific location (Advanced)
Don't search. I will choose the driver to install.
Choose this option to select the device driver from a list. Windows does not guarantee that the driver you choose will be the best match for your hardware.

Click **Have Disk** to select the path in which the driver files are located.

Insert the installation CD and click **Browse**.
Select the CD drive, double-click Driver, German and your used operating system and then select the *.inf file of the connected printer.

If you have no installation CD available, select the path in which you extracted the driver files before C:\Valentin\Driver\English
(see page 6).

Follow the instructions on the screen.

If a driver is to be installed which is not certified then a warning message appears (see page 7).

Confirm the note with **Install this driver software anyway**.

This has no reaction to the functionality of driver. However, this driver fulfills the quality conditions of Microsoft.

Click **Finish** to terminate the installation of printer driver.

Now the driver can be set up by the **Properties** dialog box.

The logo test is inapplicable for the driver versions 2.1.4. and 2.1.7 because this versions are certified.

**Installation with Add printer**

Click **Start**, **Printers and Faxes and/or Devices and Printers**.

Double click **Add Printer** and the **Add Printer Wizard** appears.

Select if you want to setup the printer as **Local printer** or as **Network printer**.

Select the **Port** you want to use.

Click **Have Disk** to select the path in which the driver files are located.

Insert the installation CD and click **Browse**.
Select the CD drive, double-click Driver, German and the used operating system and then select the *.inf file of the connected printer.

If you have no or only an old installation CD available, select the path in which you extracted the driver files before C:\Valentin\Driver\English
(see page 6)

Follow the instructions on the screen.

Confirm the message that the printer driver did not passed the Windows® logo test with **Install this driver software anyway**.
This does not have any effects on the functionality of the driver.

Click **Finish** to terminate the installation of printer driver.

Now the driver can be set up by the **Properties** dialog box.

**Installation with update**

The installation of a new driver with the update function is generally possible. However, it is not recommended because in most cases an update is not executed error-free.

## 1.3 Import - Export

The register tab **Service** provides the possibility to save or reload existing printer driver settings. In this way modifications can be made at the driver. It is always possible to access the running functions.

In the same way a running driver setting can be transferred to another driver (PC) and then imported.

# 2 Application of printer commands

## 2.1 Explanation

Printer commands are mask and parameter sets which are components of the Valentin printer language CVPL. With these commands the printer itself and the printout are controlled. The command sets can be taken from the interface manual of the used printer.

With help from direct formatting certain properties can be assigned to a label (document) without chnging the settings for printer or document before printing.

In this way it is possible e.g. to print individual pages in a document with different speed or number of copies.

Not all printers support the following specified possibilities.

**Identification - Formatting printer commands**

All following specified commands and formatting must be formatted in the command font **CVPL Command LP**. This is a normal font but it is not printed. Therefore the command can be formatted very small, so that it needs only little place in the document.

## 2.2 Formatting of text

It is possible to print individual words or entire texts rotated or inverse (white on black), provided that the text is formatted in a printer font.

Three parameters are available for the text formatting:

```
<@Rotation;Datumpoint;inverse>
```

Each parameter can be used also separately. The use of the ; sign is important to note.

When using a printer font (True Type or Bitmap) the segmentation (see page 14) is to be observed, regarding formatting. This depends from application.

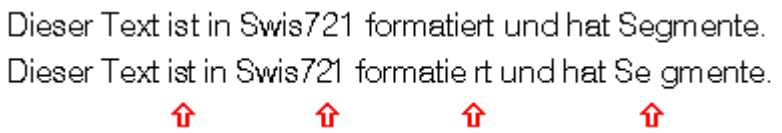
**Considering segmentation**

Application programs such as Microsoft® Word save each correction or other input separately. The outcoming is sometimes texts that are split up to illegibility. These individual text segments were passed to the printer driver. The driver must be able to differentiate if these segments are intended or if the driver must put them together.

Normally the desegmentation of vector fonts is switched off. It is active only with special commands (e.g. variables). If the desegmentation is always switched on, then a longer text is printed in only one line as long as an individual separator is found.

The segmentation has still another disadvantage because each position is transferred individually. As the printer font printed from the printer does not comply hundred percent with the presentation on screen, this results in a shift in the printout.

**Example**




In the above example are 4 segments that lead to a shift. Therefore letters are pushed together or s t r e t c h e d. This depends from the beginning of the segment. In the most unfavourable case the letters can be printed one above the other.

A desegmentation prevents this effect. With other fonts (e.g. Arial) this effect does not appear because the printer does not recognise the font and it is converted from Windows® GDI (graphic interface) in a bitmap graphic. In this case the screen display and the printout are the same.

**Desegmentation**

The desegmentation for vector fonts is normally switched on, however can be switched off in case of problems with formatting. To avoid errors in formatting individual text passages should be embedded with another font type so that beginning and end of formatting are clear.

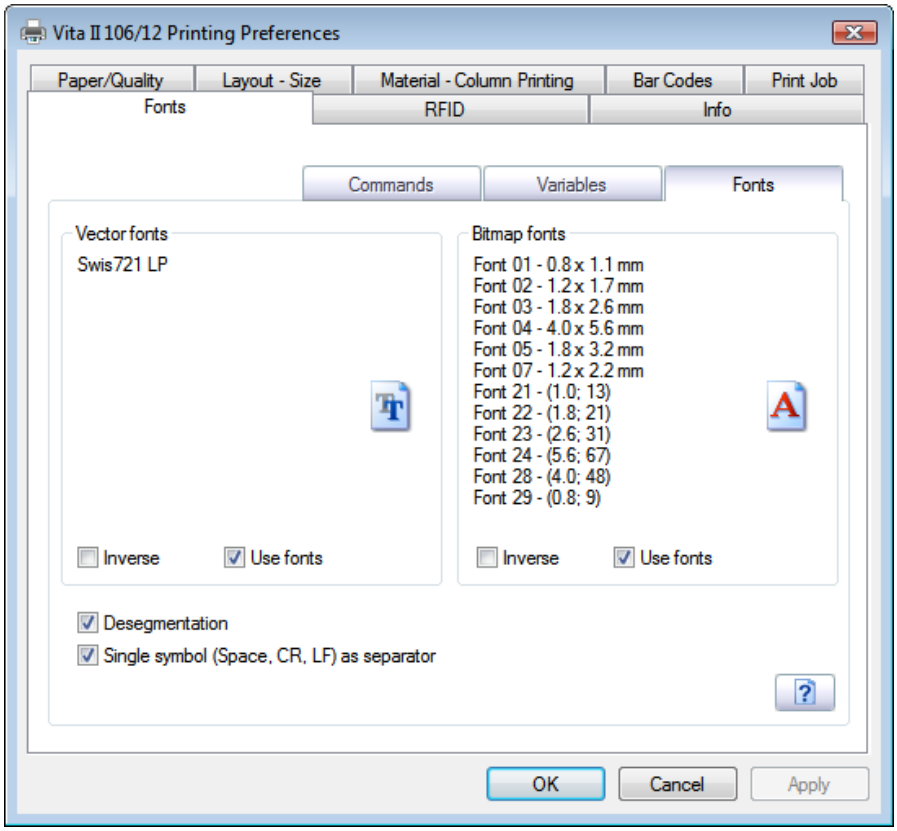
With special commands (e.g. variables) the desegmentation is generally active.

**Adjusting desegmentation**

Click the right mouse button on the used printer and select **Properties** from the context menu.

From the **General** tab click the **Printer Preferences** button.

Click the **Fonts** tab and select the smaller **Fonts** tab.




Select the check box **Desegmentation**.

It is also possible to select the **Fonts/Fonts** tab with help of the **Printer Properties** directly in the used application.

With option **Single symbol as separator** longer text can be separated by a line feed. Without this function a complete text block would be printed in one line only. With some applications e.g. Word individual blanks can be interpreted as separators by segmentation.

**Inverse printing**

Click the right mouse button on the used printer and select **Properties** from the context menu.

From the **General** tab click the **Printer Preferences** button.

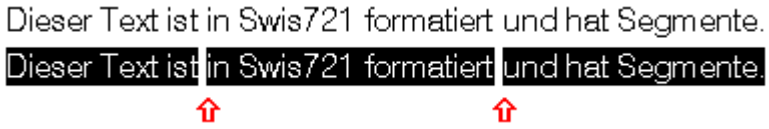
Click the **Fonts** tab and select the smaller **Fonts** tab.

Select the check box **Inverse**.

It is also possible to select the **Fonts/Fonts** tab with help of the **Printer Properties** directly in the used application.

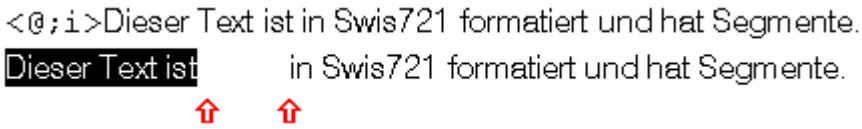
With this option the inverse printing is activated for all vector and bitmap fonts at the same time. According to this, each printer font in the document is printed inversely.

**Example 1**




The upper line shows the indication in the application and/or on the screen and the second line shows the printout. In this example the individual segments can be recognised particularly.

**Example 2**




The upper line was formatted in CVPL Command LP and indicates the screen display. The second line shows the printing result. In this example individual text passages were formatted.
In this case the large gap is noticeable and that the complete text is not printed inverse. In addition, this error shows the function of the command and the consequences of segmentation.

<@;i> In order to activate the inverse printing, the command must be formatted in CVPL Command LP.
Afterwards the command can be formatted in a very small character size e.g. 1 or 2 so that the formatted text is not disturbed.

**Rotating text**

For rotation of text the possibilities are 90°, 180° and 270°.

With the command <@270>, the following text is rotated by 270°.

These commands can be combined also with variables.

| **Command** | **Printout** |
|---|---|
| `<@270>=CN(0;0;3;+1;1)050` | 050 |

**Datumpoint/ Initial point**

The previous example shows that rotation was effected at the point at left bottom. In order to place the printout in the same line as the command, the datumpoint of rotation can be adapted.

| **Command** | **Printout** |
|---|---|
| `<@270;8>=CN(0;0;3;+1;1)050` | 050 |

In this example the figure was rotated and printed on the correct height.

For the selection of datumpoint the possibilities are:

| | | |
|---|---|---|
| 1 = left top | 2 = centre top | 3 = right top |
| 4 = left centre | 5 = centre | 6 = right centre |
| 7 = left bottom | 8 = centre bottom | 9 = right bottom |

The function of formatting must be tested in individual cases.

In the same way all printer fonts (also bar codes) can be formatted.

Caused by the general activation of desegmentation the complete text of the previous example is printed inverse and without interruption.

## 2.3 Printer commands

Printer commands are all commands that affect the printer and the printing behaviour.

**Identification - Formatting printer commands**

All following specified commands and formatting must be formatted in the command font **CVPL Command LP**. This is a normal font but it is not printed. Therefore the command can be formatted very small, so that it needs only little place in the document.

**Optional commands**

In principle all mask and parameter sets can be sent directly to the printer.

**Example**

```
FCCO--r0010570-
AM[1]559;9787;0;1;0;22;1;1;2
BM[1]Now the RFID user data follow
```

With the integrated position and object recognition it is possible to use a short version for mask sets, i.e. without indication of xy-coordinates.

```
AM[]0;1;0;22;1;1;2
BM[]Now the RFID user data follow
```

The definition of the different mask and parameter sets can be learned from the interface description.

**Pre-defined short commands**

For an easier use and definition of the above described commands different short commands were integrated in the driver.

These commands are applicable with the following schema:

**Syntax**

```
<!BEFEHL>Parameter
```

**Starting from version 2.1.3 the following commands are integrated:**

```
<!SPEED>
<!CONTRAST>
<!LENGTH>
<!WIDTH>
<!COPIES>
<!SINGLECUT>
<!FORMFEED>
<!ROTATE>
<!ENDCUT>
<!NO_FBC>     (from beginning with version 2.1.7)
```

**Singlecut / Formfeed**

Directly executed commands such as SINGLECUT and FORMFEED must be placed at the position in the document where they are needed. If a cut should be activated after the second page, then the command must be placed on page 3.

**Endcut**

In addition, ENDCUT can be used alternatively. Hereby a single cut is only sent after the 'print start' command.

**Rotate**

With ROTATE it is possible to print out printer fonts on a single page rotated (also in a multiple page document).
The command has no effect on other fonts because the printout is controlled from the application.

**Length / Width**

If the wrong form size should be sent nevertheless to the printer (e.g. in Client Server environments) despite all corrective measures of the driver, the correct size can be defined with the commands LENGTH and WIDTH. This command can be used e.g. when printing on continuous material and the labels to be printed are differently long.

**Copies**

With COPIES it is possible to print an individual page in a multi page document several times. If the following pages should be printed one, then the number of copies must be again reset with the command <!COPIES>1.

**Example**

on page 1: <!COPIES>1
on page 2: <!COPIES>2
on page 3: <!COPIES>1

In this case the first and third page is printed once, the second page is printed twice.

**Speed / Contrast**

With SPEED and CONTRAST individual documents can be printed with modified speed and/or contrast.

**Example**

A certain paper size is inserted in the printer and it is always printed with the default setting. For specified labels the print speed is to be reduced and generally 2 copies are to be printed.

For this, the following commands must be available at an optional place in the document:

<!SPEED>70
<!COPIES>2

The driver recognises these commands and uses the appropriate parameter sets.

**Advantages of short commands**

As the previous example shows, the printer settings can be set individually for the document. Knowledge for the writing of the parameter set is not necessary. A short command in clear text is sufficient and is understandable for each user.

**Restrictions**

The functionality of commands must be tested in individual cases. Not each combination is possible.

# 3 Application of 2D bar codes

## 3.1 Explanation

2D bar codes can save a lot of information on relatively small area. Some 2D bar codes have an integrated error correction, so that the information can be still read if some of the 2D bar code is damaged.

2D bar codes can be read with image processing systems. Some mobile phones integrated such programs.

Not all printers support the following specified possibilities.

**Formatting 2D bar codes**

All following specified commands inclusive the text to code must be formatted in the command font **CVPL Command LP**. It is a normal font, however, it is not printed. Therefore the command can be formatted very small, so that it needs only little place in the document.

**Integrated 2D bar codes**

```
<#PDF417>Text
<#MAXICODE>Text
<#DATAMATRIX>Text
<#CODABLOCK>Text
<#GS1>Number|Text
<#QR>Text
<#ITF14>Number
```

The ITF14 is no 2D bar code but it is processed from the driver in the same way and is therefore displayed in the same list.

**Example**

**Command**

```
<#PDF417>Das ist ein PDF417
Barcode
```

**Printout**

The bar code contents is still available and readable in the document, therefore it can be corrected at any time.

**Modifying parameters**

According to the indications in the interface description each parameter can be set. The values of the used parameter must be taken from the corresponding documentation.

The parameters can be changed when using the following syntax:

```
<#CODE;Parameter:Value;>Text
```

It is possible to use each parameter that is described in the interface manual.

**Example**

```
<#PDF417;s:15>Das ist ein Text
```

**Print position**

The initial point of 2D bar code matches with the position of command, therefore the 2D bar code is printed over the text which is placed in the lines above. For this reason it is meaningful to change the datum point by means of the parameter dp:

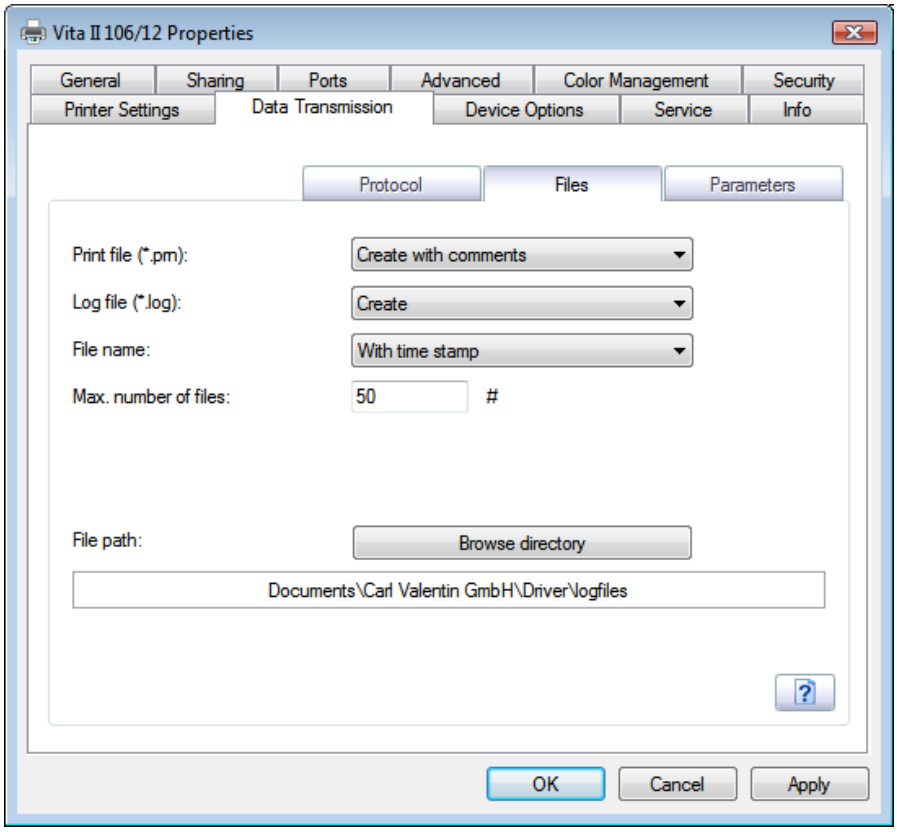
```
<#PDF417;s:15;dp:1;>This is a text
```

**Error analysis**

If a 2D bar code is printed incomplete and incorrect, by means of the log file it can be determined if the complete text was used.

In order to setup the prn and log files, click the right mouse button on the used printer and select from the context menu **Properties**.

Click the **Data transmission** tab and select the **Files** tab.



# 4 Application of variables

## 4.1 Definitions

**Variable**

According to the interface description, counters (consecutive numbering), time and date and other printer functions can be created by means of variables.
With help from a certain code the printer can print the appropriate values automatically.

**Object**

The driver differentiates between general text (e.g. Arial), text in a printer font (e.g. Baskerville) and graphics.

Text formatted in a printer font is recognised from the driver as an object. Objects can be processed from the driver itself.

Other text is returned to the ® GDI (graphic interface). Thereupon the GDI creates a graphic of the text that is to print and passes it again to the driver.

**Desegmentation**

Applications such as Microsoft ® Word divide the entered text into individual segments (see page 14). The more corrections were done the more segments are created.

When using a printer font then individual objects arise.

The desegmentation is a block building which can combine again the individual segments.

**Used fonts**

| Vector fonts | Bitmap fonts |
|---|---|
| Swis721 (bold)<br>Swis721 (bold italic)<br>Swis721<br>Swis721 (italic)<br>Swis721 light<br>Swis721 light (italic)<br>Baskerville<br>Baskerville (italic)<br>Brush Script<br>Brush Script (italic) | Font 01 (0,8 x 1,1 mm)<br>Font 02 (1,2 x 1,7 mm)<br>Font 03 (1,8 x 2,6 mm)<br>Font 04 (4,0 x 5,6 mm)<br>Font 05 (1,8 x 3,2 mm)<br>Font 07 (1,2 x 2,2 mm)<br>Font 21 (1,0; 13)<br>Font 22 (1,8; 21)<br>Font 23 (2,6; 31)<br>Font 24 (5,6; 67)<br>Font 28 (4,0; 48)<br>Font 29 (0,8; 9) |

**Variable structure**

The variable structure can be learned from the appropriate interface description.

**Function of driver**

The driver can combine as many as desired individual text segments into a functioning command, so that this is recognised by the printer.

## 4.2 Counter

**Example**

=CN(0;0;4;+1;1)0000 formatted in Swis721

With this entry all copies of the document were provided with a consecutive four-digit number.

Blanks must be avoided, otherwise undefined printouts or an error messages appear.

| **Input** | **Printout** | | | |
|---|---|---|---|---|
| With blank after CN | 123 | 123 | 123 | etc. |
| Without blank after CN | 0000 | 0001 | 0002 | etc. |

## 4.3 Date/time

The indication of the current time and/or date operates according to the same principle.

With the definition of date it is possible to add offset values.

**Example**

Date according to RTC: 30.01.09

| **Input** | **Printout** |
|---|---|
| =CL(0;0;0)<DD.MO.YY> | 30.01.09 |
| =CL(1;1;0)<DD.MO.YY> | 31.01.09 |

With the definition of the date offset of values can be added.

## 4.4 Using in document

For the new algorithm on desegmentation (block construction) it is important that certain rules are observed. The algorithm can operate correctly only if the entered text is recognised.

**Example**

**Input with tabulator**

=CL(0;0;0)<DD.MO.YY>, =CL(0;0;0)<HH:MI:SS>

**Printout**

30.01.09, 11:41:25

The single tabulator between both variables serves the driver as end sign. With the new = sign starts a new block construction.

If time is not printed, both variables must be separated by using another font type.

By the clear separation from each other the driver must start a new block construction.

**Input with blank**

=CL(0;0;0)<DD.MO.YY>, =CL(0;0;0)<HH:MI:SS>

**Printout**

30.01.9Y >, =CL(0;0;0)<HH:MI:SS>

**Insight into the log file**

Swiss 721 Light LP: **=**
Swiss 721 Light LP: **CL**
Swiss 721 Light LP: **(0;0;0)<DD.MO.YY**
Swiss 721 Light LP: **>, =**
Swiss 721 Light LP: **CL**
Swiss 721 Light LP: **(0;0;0)<**
Swiss 721 Light LP: **H**
Swiss 721 Light LP: **H**
Swiss 721 Light LP: :**MI:SS**
Swiss 721 Light LP: **>**

A strong segmentation is recognisable. The driver converts the following:

**Insight into the prn file**

AM[1]2210;9796;0;4;0;5;254;254;0
BM[1]**=CL(0;0;0)<DD.MO.YY**
AM[2]2210;6155;0;4;0;5;254;254;0
BM[2]**>, =CL(0;0;0)<HH:MI:SS>**

The driver searches for = as new start (starting from version 2.1.4 the = sign must be places as the first indication in the segment).

**Printout**

30.01.09, =CL(0;0;0)12:38:21

**prn file**

AM[1]2210;9796;0;4;0;5;254;254;0

BM[1]=CL(0;0;0)<DD.MO.YY>, =CL(0;0;0)<HH:MI:SS>

The examples show that the second command =CL(… can be removed and simplified.

The following command can be used.

=CL(0;0;0)<DD.MO.YY>, <HH:MI:SS>

This would lead directly to the correct result.

Moreover the example shows that with incorrect printout an appropriate formatting must be used.

If the blank is formatted at the beginning in Arial then the printout is immediately correct.

## 4.5 Using wildcards

The previous chapters (4.2 - 4.4) described the use of individual variables. However, in many cases the space in document is not sufficient to use these long definitions.

The use of wildcards simplifies the use of variables. If the driver recognises a wildcard its text is replaced from driver with the appropriate variable definition.

A further advantage for wildcards is that their function is more easily recognisable by the describing name, i.e. <§time> can be recognised more easy as time indication as =CL(0;0;0)<HH:MI:SS>.

**Example**

In smaller labels the definition of the time variable cannot be used because of a lack of space.

| **Variable** | **Printout** |
|---|---|
| =CL(0;0;0)<HH:MI:SS> | 12:38:21 |

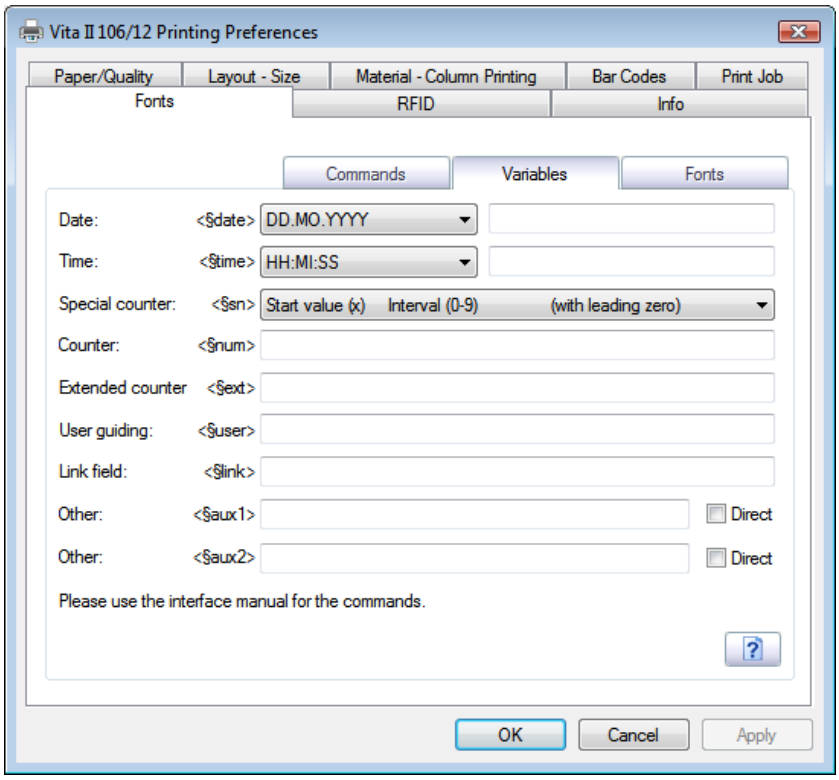
| **Wildcard** | **Printout** |
|---|---|
| <§time> | 12:38:21 |

## 4.6 Defining wildcards

Click the right mouse button on the used printer and select **Properties** from the context menu.

From the **General** tab click the **Printing Preferences** button.

Select the **Fonts** tab and then click the **Variables** tab.




Each wildcard can be defined separately. If a wildcard is recognised in any document this definition is used.

The definition of wildcards is valid for all documents which are printed. If different functions of a variable (e.g. two different time formats) are to be used these variables must be defined directly in the document see chapter 4.2 - 4.4).

Only the first three wildcards (<§date>, <§time> and <§sn>) are replaced by their appropriate variables. All other wildcards are freely definable but should be used according to their names

Wildcard must be defined in a printer font.

**Date / Time**

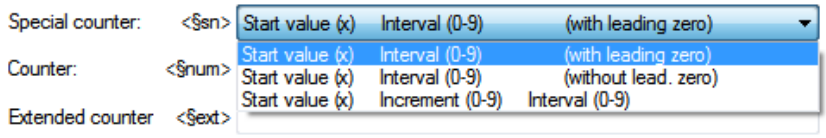
With the wildcards for date and time there are some predefined formats which can be selected in the appropriate selection fields. If other formats are to be printed the necessary formatting is to be taken from the corresponding interface manual of the used printer and to be adapted in the appropriate text field.

**Predefined Counter**

For the definition of counters are different possibilities available.

**Example**

<§sn>102


Special counter: <§sn>
Start value (x) Interval (0-9) (with leading zero)
Counter: <§num>
Start value (x) Interval (0-9) (with leading zero)
Start value (x) Interval (0-9) (without lead. zero)
Extended counter <§ext>
Start value (x) Increment (0-9) Interval (0-9)

**Option 1**

**Interval with leading zeros (CN)**

| | |
|---|---|
| Start value | 10 |
| Interval | 2 |

**Printout with 6 copies**

10, 10, 11, 11, 12, 12

**Printout with 250 copies**

010, 010, 011, 011, 012 etc. 099, 099, 100, 100, 101, 101 etc.

**Option 2**

**Interval without leading zeros (CC)**

| | |
|---|---|
| Start value | 10 |
| Interval | 2 |

**Printout with 250 copies**

10, 10, 11, 11, 12 etc. 99, 99, 100, 100, 101, 101 etc.

**Option 3**

<§sn>1032

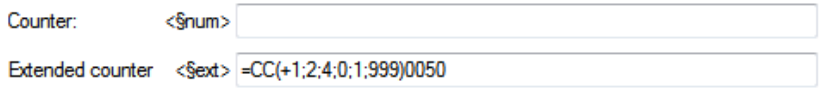
| | |
|---|---|
| Start value | 10 |
| Step width | 3 |
| Interval | 2 |
| Increment | 3 |

**Printout with 6 copies**

10, 10, 13, 13, 16, 16

**Counter freely to define** <§num> or <§ext>

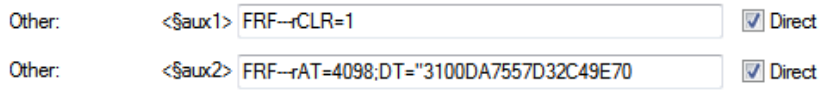
Hereby no start values etc. must be indicated. The definition is effected by the driver only. In the input field of the driver <§num> the complete counter can be indicated.




If the driver recognises the text <§num> or <§ext>, then the definition which is indicated in the input field is inserted instead.

**Direct printout**

With the 'aux' fields it is possible to define as much as desired commands. Select the **direct** option (see below) to include the command directly in the data stream which is to be transferred to the printer.




In this case no mask set was created.

**Link field**

The definition of a link field can be effected with a wildcard <§link>. The object numbers must be taken from the print file.

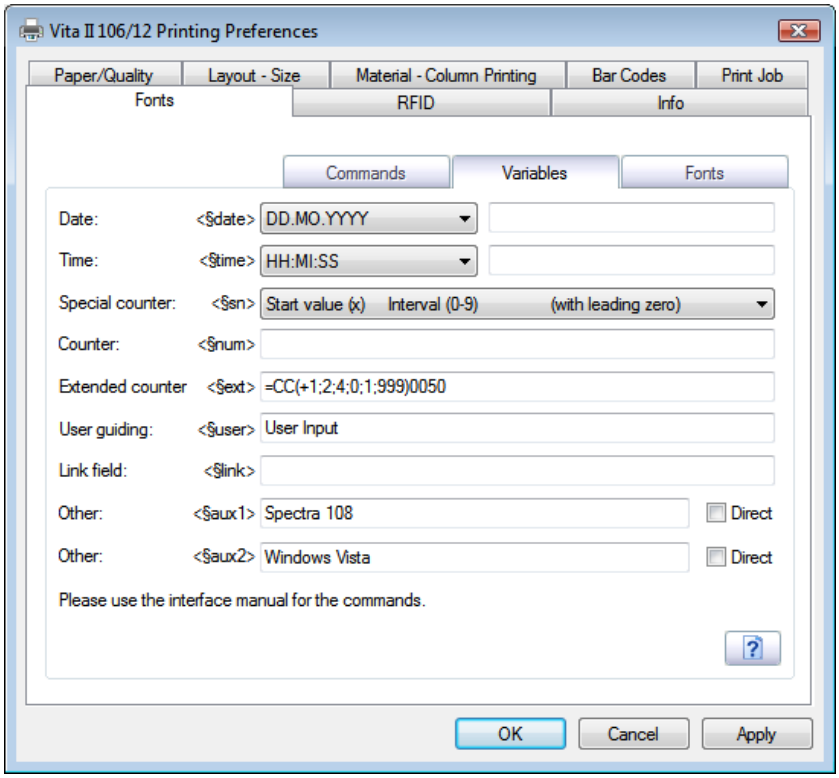
Over the fields “aux” arbitrary instructions can be defined. By selection of the option “directly”, the instructions can be merged directly into the data stream, which will transfer to the printer.

**Customised entry**

With the cutomised entry it is possible to control inputs at the printer. In the printer display appears a prompt.

## 4.7 Label example

<§date> <§time>
<§num> <§ext>
<§aux1> <§user>
<§aux2> <§sn>9852
<§link>




03.03.2009 08:50:03
0000 50
Spectra 108 User input
Windows Vista 985
03.03.2009 08:50:03 0000

With empty fields the driver uses partly pre-defined examples so that the function can be noticed.

The link field <§link> is the variable =SC(1;" ";2;" ";3) stored. The link field places the objects 1,2 and 3 one after the other and creates a new object which is printed.

# 5 Error handling

There are some errors which result from wrong settings in the driver. The most common error is the e.g. the use of a wrong format.

## 5.1 Recognising problems

**From document to printer**

The driver informs the application which (bitmap) fonts, form sizes etc. can be processed.

Before the application is started and/or a new document is created, the driver must be selected. Only if the correct driver was selected, the correct data can be used and/or processed.

In order to print a document, the application transfers data to the driver. In this way the readable text, the used font type, graphics etc. is provided to the driver for further processing.

If a printer font (e.g. Swis721) is recognised from the driver then a mask set is created. TrueType fonts (e.g. Arial) are not recognised from the driver and are therefore returned to the application.

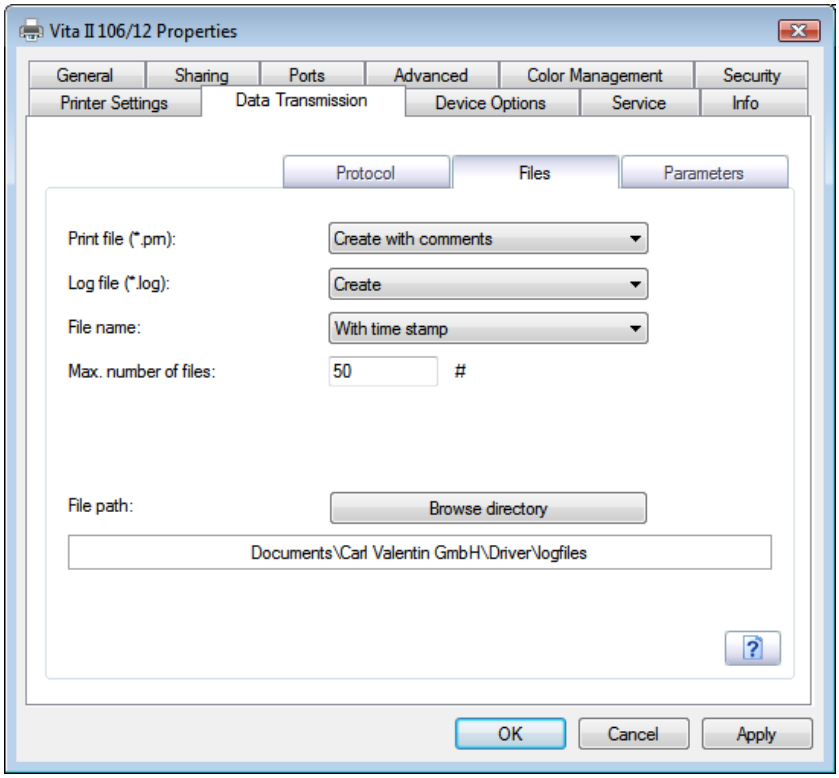
Unknown texts are converted from the GDI (Windows® graphic interface) in graphic data and sent again to the driver.

The driver bundles all data and creates a data stream which is sent to the spooler (Windows® component). The spooler supervises the individual print orders and sends it finally to the printer.

## 5.2 Protocol files

Click the right mouse button on the used printer and select **Properties** from the context menu.

Click the **Data transmission** tab and then select the **Files** tab.




From the print file (*.prn) you will see which print data were created. When using time stamps it is possible to produce successively several print files and to compare them.

The log file (*.log) provides information about settings of the driver, texts and fonts which derive from the application and which label forms are used.

**Log file**

The log file contains all driver settings. The most important section is at the end of the file.

```
 [START_DOC]
Swiss 721 LP: Ein Text in Swis721
Swiss 721 LP:
Arial: Ein anderer Text in Arial
Arial:
Calibri:

 [END_DOC]

Public DEVMODE (CMDID_END_JOB) at: Date: 04.02.2009 Time: 12:13:27,866
  DeviceName   =
  Orientation  = 1 - valid
  PaperSize    = 141 - valid
  PaperLength  = 300 - not used
  PaperWidth   = 1000 - not used
  FormName     = Etikett 100 x 30 mm - valid

 [END_JOB]
```

The clear text indicates the text which is to be printed.
Example: 'Ein Text in Swis721' and 'Ein anderer Text in Arial'.

Moreover the paper form is indicated.
Example: 100 x 30 mm.

**Print file**

The following extract shows the appropriate prn file. The label format 100 x 30 mm is also recognisable. In addition it is obvious that from the text formatted in printer font Swis721 a mask set was created.

The apparently unsystematic character sequence SOH NUL ETB ESC B NUL etc. belongs to the bitmap graphic that was created from the 'Arial' text.

```
SOHFCCL--r0003000-ETB  // LabelLength 30.0 mm
SOHFCCM--r00200---ETB  // GapLength 2.0 mm
SOHFCCO--r0010000-ETB  // Labelwidth 100.0 mm
SOHFCCJ--r0-------ETB  // Label alignment - left
SOH#0500001B0004AYSOHNULETBESCBNULrNULESCASOHSINULRSNULS
   // END_PAGE
ESCC
SOHAM[1]364;9796;0;4;0;3;279;279;0ETB
SOHBM[1]Ein Text in Swis721ETB
SOHFBAA--r1ETB         // Objects(Fields) 1
SOHFBC---r1-------ETB  // Printstart unsortiert
   // END_DOC
SOHFCDD--r1-------ETB  // Cutter Singlecut
   // END_JOB
```

The prn file shows the data stream which the driver sends to the spooler and/or to the printer. It is composed of parameter sets (printer and document settings), graphic data, mask sets (print objects) and the actual print start.

## 5.3 Possible errors

**Missing text**

If a text in the log file is not displayed then it must be examined if the text is placed possibly in the not printable range. The format and margins must be examined in the application.

**Unreadable text**

Individual programs partly transfer no clear ASCII code (readable characters) to the driver.
Not printable special characters are recognised and lead to errors with printer fonts.

**Example of a log file**

```
118 Arial: 7HVW<ETX>(&8
119 Arial: 6HULDO<ETX>1f
120 Arial: 3DUW<ETX>1f
121 Arial: 2(0<ETX>1f
122 Arial: <DC4><NAK><SYN><ETB><CAN><EM>
123 Arial: +:<ETX>&RQWURO
124 Arial: +:<ETX>,GHQW
125 Arial: 2(0<ETX>1DPH
126 Arial: H
```

If the use of printer fonts is switched off in this example, the application must generate the printout itself and the GDI must create a graphic. These graphic data are printed correctly.

If a printer font should be used nevertheless, the error must be searched in the application itself and then removed.

**Text with shift**

This problem concerns printer fonts only.

The segmentation, that is created at each correction e.g. in Microsoft® Word, result in text blocks. Each text block has its own starting position (x-y coordinates).

Because of the fact that the screen display cannot be copied 100% when using printer fonts, rough passages result from smallest differences in character sizes. Letters can be pushed together or stretched. In this case the position of the individual segments is responsible.

In order to avoid the shift from the Fonts tab the desegmentation is to be switched on or the use of printer fonts must be switched off.

**Wrong format**

For the general use of form sizes there is no guideline. The settings of system and application should be in accordance. An already created form which is to be print on another computer must be again set up there.

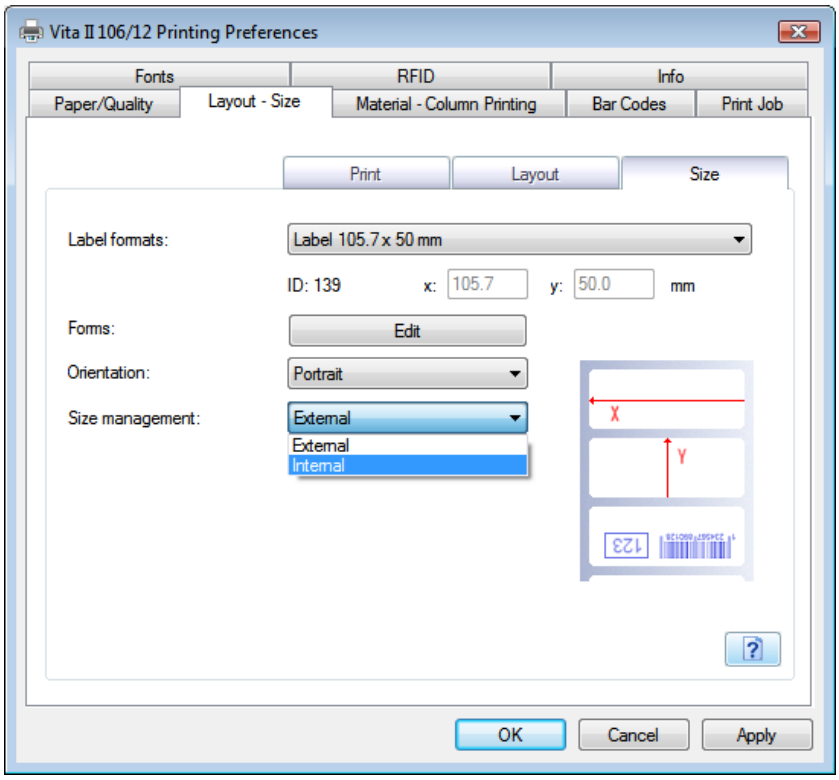
**Wrong settings**

In order that the settings in a program that are made in the **Printer Preferences** are used, the **Size management** must be adjusted accordingly.

Click the right mouse button on the used printer and select **Properties** from the context menu.

From the **General** tab click the **Printer Preferences** button.

Select the **Layout - Size tab** and then the **Layout** tab.




**Extern - Application** uses the label format and orientation from the application and **Intern - Driver** uses the the values from driver.

Even if the application determines the label format the size should be entered in driver in order to avoid error messages of the application.

The correct setting depends on the used program and must be tested in individual cases.

If the orientation is set to **Portrait - Landscape** no new label format can be selected.

# 6 Index


## 2

2D bar codes
- Error analysis ... 22
- Explanation ... 21
- Integrated 2D bar codes ... 21
- Parameter, modify ... 22
- Print position ... 22

## D

Driver files, extracting ... 6
Driver, delete
- Windows® Vista ... 5
- Windows® XP ... 5

## E

Error correction ... 33
Error handling ... 29, 30, 31, 32
Extracting, driver files ... 6

## I

Installation
- Add printer ... 12
- Plug & Play ... 10
- Preparations ... 5
- Setup ... 8
- Update ... 12

## P

Presetting, driver ... 12
Printer commands
- Datumpoint/initial point ... 17
- Desegmentation ... 15
- Explanation ... 13
- Formatting ... 13
- Identification ... 13
- Inverse printing ... 16
- Pre-defined short commands ... 18
- Rotating text ... 17
- Segmentation ... 14

## V

Variables
- Counter ... 22
- Date/time ... 22
- Definitions ... 21
- Using in document ... 23

## W

Wildcards (variables)
- Defining ... 25
- Using ... 24